

Software as a bridge between theory and practice in life cycle assessment

Reinout Heijungs and Jeroen Guinée

Centre of Environmental Science, PO Box 9518, 2300 RA Leiden, The Netherlands

Received 17 July 1993; revised 10 January 1994

Three mutually dependent elements are required for the application of life cycle assessment: methodology, data and software. Obviously, the design of software is determined by the methodology and the type of data available. Conversely, the development of software dictates the way in which data should be collected and recorded, and improves the theoretical framework, as it forces one to state the principles clearly and unambiguously. The influence of the development of software on both data and methodology is addressed and illustrated by examples, with reference to two key terms: transparency and explicitness. Three types of influence are distinguished: the design of a protocol, the formulation in terms of recipes, and the presentation of data.

Keywords: life cycle assessment; software; methodology

Developments with respect to life cycle assessment

Environmental policy is becoming increasingly product-oriented. The main reason for this is that products are the core of all economic activities. A sound product-oriented policy poses a high demand with respect to information on the environmental performance of products, including related industrial processes. Life cycle assessment (LCA) is acquiring a prominent position among the instruments to support product-oriented environmental policy.

This paper will concentrate on the relation between software development and methodology, and between software development and the compilation of data. Besides international coordination, for which SETAC¹ is the leading organization, there are many research activities in the field of LCA. These can be subdivided into three areas: methodology²⁻⁶, software⁷⁻⁹ and data^{8,10-12}. The development of methodology for LCA is highly theoretical, whereas the collection of data has a direct connection with practice. Software takes a position in between: it contains the formalized methodology in a way that is accessible to the data, with its practical limitations. The development of software increases the practical usability of the methodology and the suitability of data within the theoretical framework. Software may thus act as a bridge between theory and practice.

The influence of the development of software on methodology will be elucidated in the following sections. One may characterize this influence by the words *transparency* and *explicitness*. The use of these

terms may seem trivial. However, examples which show that transparency and explicitness are often difficult to achieve, and that their absence may lead to misunderstandings, will illustrate their significance.

Methodology

Transparency

LCA could be defined as an input-output relation, i.e. a function that maps data onto a result. In that case, the methodology for LCA is defined as a black box. This functional definition of the methodology for LCA is not transparent. When two black-box LCAs give a different result, it is not clear what has caused this¹³⁻¹⁵. Since opinions differ, for example, on the correct procedure for allocation of multiple processes, this may be the reason for different answers. But it may equally well be due to other methodological aspects, such as the weighting of toxic emissions.

Many widely used computer languages are imperative, i.e. the source code consists of a sequence of tasks. The operations to be performed in LCA are thus broken into a sequence of elementary operations. A consequence of this is an enhanced transparency: the black-box LCA is now divided into a number of unit steps.

There is a natural sequence of elementary steps in LCA. Although the components — which are an aggregate of related steps — have been the subject of some controversy^{2,16}, there seems to be more agreement on the steps. *Table 1* gives an example of a stepwise procedure for LCA. Of course, one may debate the appropriateness of certain steps, and the

Table 1 An example of a proposal for a number of elementary steps, which fit in the current framework (components according to ref. 1; steps taken from ref. 5)

| Component | Step | Question which is answered | |
|--|-------------------------------------|--|--|
| Goal definition and scoping | Determining the application | Why, for whom and by whom? | |
| | Determining the depth of the study | How long and expensive in relation to the application? | |
| | Defining the subject of the study | What and how much? (functional unit) | |
| Inventory analysis | Drawing up the process tree | Which processes and which not? (system boundaries) | |
| | Entering the process data | How do the individual processes look like? | |
| | Application of the allocation rules | To what extent are emissions, etc. to be attributed to each of the coproducts? | |
| | Creating the inventory table | Which emissions and resource extractions result from the functional unit? | |
| Impact assessment | Classification | What are the endpoints of the impact assessment? | |
| | Characterization | Definition of the classification factors | What is the contribution of a unit emission, etc. to the endpoints defined? |
| | | Creating the environmental profile | What are the contributions per functional unit to the endpoints defined? |
| | | Normalization of the effect scores | What are the contributions per functional unit relative to the annual world's problems? |
| | Valuation | Evaluation of the environmental profile | What does a subjective weighting of the problem types imply for a ranking of alternatives? |
| Evaluation of the reliability and validity | | How rigorous is the ranking with respect to uncertainties in data and assumptions? | |
| Improvement assessment | Dominance analysis | Can one identify bottlenecks? | |
| | Marginal analysis | Which modifications in process data affect the results strongly? | |

implementation of any of these steps may be the subject of discussion. The advantage, however, is that these discussions can be extremely pertinent. Separation into steps makes it possible to concentrate on the things that really matter.

Explicitness

In addition to increased transparency, development of software requires explicitness. The use of an imperative computer language implies the formulation of clear and unambiguous statements. The elementary steps distinguished within the encompassing black-box LCA must be specified by supplying an explicit recipe. Recipes, though obviously needed, are often lacking.

There are various levels for such a recipe. Compare the following equivalent statements, which could be found in a report on impact assessment:

- 1. The emissions of greenhouse gases will be assessed using global warming potentials.
- 2. The mass of each greenhouse gas will be multiplied with its corresponding global warming potential, after which the results will be added to yield one score for global warming.
- 3. The score for global warming, GW, is given by:

$$GW = \sum_{subs} GWP_{subs} \times m_{subs}$$

where *subs* = an index which applies to all greenhouse gases, *GWP_{subs}* = the global warming

potential of greenhouse gas *subs*, and *m_{subs}* = the emitted mass of greenhouse gas *subs*.

4. GW = 0
 DO 10 ISUBS = 1, NSUBS
 GW = GW + GWP (ISUBS) * M(ISUBS)
10 CONTINUE

Formulation 1 is not explicit. A novice in the field or a computer specialist designing software for LCA may be unaware of the fact that one should multiply the global warming potential and not divide it. Formulation 4 is an extreme example of an explicit formulation. This is, of course, not the intention: the development of reliable and user-friendly software remains a specialized work, not to be covered by LCA specialists. Moreover, though explicit it is not very clear. The two middle formulations, 2 and 3, offer a good compromise: they are explicit, and not too specific. It remains a matter of personal opinion, skill, purpose and target audience which of the two to prefer.

Examples of steps where an explicit recipe is required are:

- allocation: not just 'allocation was done on a mass basis', but which masses (e.g. excluding or including adhesive water¹⁷), and what has been allocated to what;
- inventory table: specify how the proportion of the processes involved has been calculated, including the way to handle loops;
- characterization: see the GWP example above;

- normalization: specify if and how normalization was done, including which data were used;
- valuation: specify weighting procedure and the way weighting factors were derived.

By giving explicit formulations of the activities inside a step, a very direct translation to software is possible. This reduces the risks of misinterpretations by the computer specialist who undertakes this task.

Explicitness also facilitates discussions between different schools of methodology. People can understand each other's opinions faster and better. Discussions can concentrate on why choices are made in a certain way, without dwelling on what choices are made. A judgement by peer review is inconceivable without explicit statements, and the reproducibility of the results is enhanced.

A last argument is that explicit formulations offer new and often unexpected possibilities. This argument applies in particular to the mathematical formulation 3. With some mathematical education, explicit formulations may be investigated to derive certain properties, or several equations may be combined to reveal 'hidden' results. An example of this enhanced power can be found in ref. 18, where a method to calculate the inventory table was developed; it proved possible to cast this method into an explicit formula, which was further analysed to yield a sensitivity analysis and a method to find options for improvement. These results are inconceivable with non-explicit statements such as 1, and almost impossible to derive with non-symbolic formulations such as 2.

Data

So far, emphasis has been on the impact of software on the theoretical part of LCA: the methodology. Now we will address the practice: the influence of software on data.

A large amount of data is required for the implementation of an LCA. A key word in this context is again *explicitness*. Whereas in every-day life many things are tacitly assumed, a binary computer has to be fully instructed. An example may illustrate this. Many reports on LCA contain process data in the form of tables. One frequently encounters tables with a caption 'emissions for 1 kg PVC'. An LCA practitioner knows that this means 'emissions associated with the production of 1 kg PVC'. But the phrase is not unambiguous: it could well mean 'emissions associated with the transportation of 1 kg PVC' or 'emissions associated with the incineration of 1 kg PVC'.

Other arguments may strengthen this point, for example, with respect to waste. A distinction between waste streams within the economy and to the environment is important. An effluent discharge may occur directly to the surface water. In that case the waste flow is emitted to the environment. In other cases, effluents are discharged to a municipal sewage system. This means that the waste remains somewhat longer in the economy while it is treated by a sewage

treatment plant, and that the treatment plant is part of the process tree. Hence the system boundary between economy and environment has consequences for the format.

Another argument stems from the allocation problem. Many processes produce other minor streams besides the main product or material. Some of these are considered as waste, but others represent some valuable input to other processes. The allocation problem deals with the question of what to allocate to what. The minor flows that are waste streams should be allocated to the various usable products and materials, i.e. to the main and minor flows that are of value. It should be clear from a process specification which flows are of value, and which are not.

A last example relates to processes (such as 'Zellstoffherstellung'¹¹) that both require and emit SO₂. The first item is an economic input, the second an environmental output.

These considerations force one to structure process data in a systematic way, for which purpose a basic template can be designed (*Figure 1*). To summarize, the basic distinctions of origin, destination or properties of the flows are: input/output; economy/environment; and valuable/non-valuable. This template may serve as a guideline in the construction of a more elaborate format. Aspects that are also of interest for the possibilities of endorsement of process data are the choice of names for materials, substances, etc., the choice of units, the decimal representation, the distinction between missing entries and zero entries, etc.

The nomenclature of extracted resources and emitted substances is a problem that concerns both inventory analysis and impact assessment. Inventory analysis and impact assessment should 'match' with respect to substance names. Uniformity, though desirable, is often not yet achieved⁵ and could be realized by conforming to databases with substance properties. However, there will always remain problems, for instance when inventory data are less detailed than required for impact assessment. An example of this is knowledge of emissions as halons, whereas the ODPs make a distinction between halon 1301, halon 1302, etc.

Concerning the names of products, materials, services and processes, there are other problems. The problem of missing values, for example, can be illustrated as follows. Assume that global warming is characterized by GWPs. IPCC does not give a GWP for phenol. We may assume that this means that phenol has a zero GWP. Next assume that we have a database of processes containing data on the generation of electricity, and that there is no information on emissions of cadmium. What does this mean? Is there absolutely no emission of cadmium? Or is it below the detection limit? Or just unknown? Again we assume in many cases that it is absent, but we must help software interpreting. So why not be directly explicit?

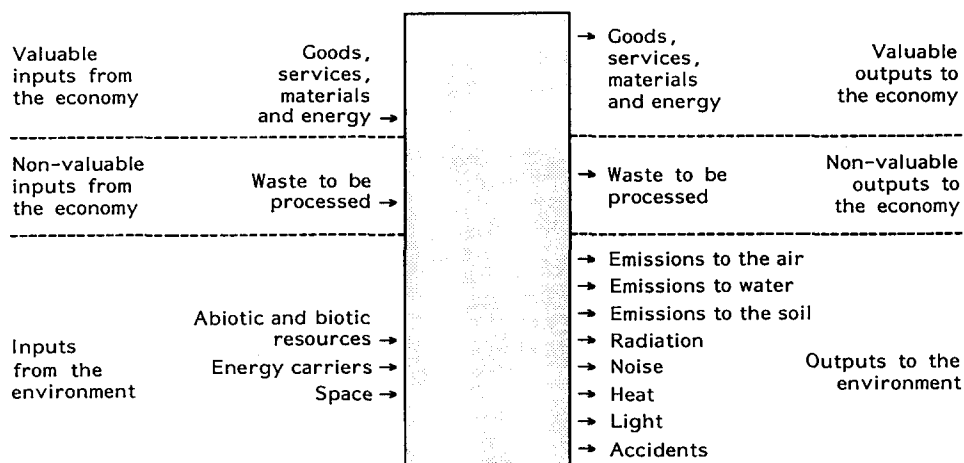


Figure 1 A basic template giving a clear structure for the presentation of process data

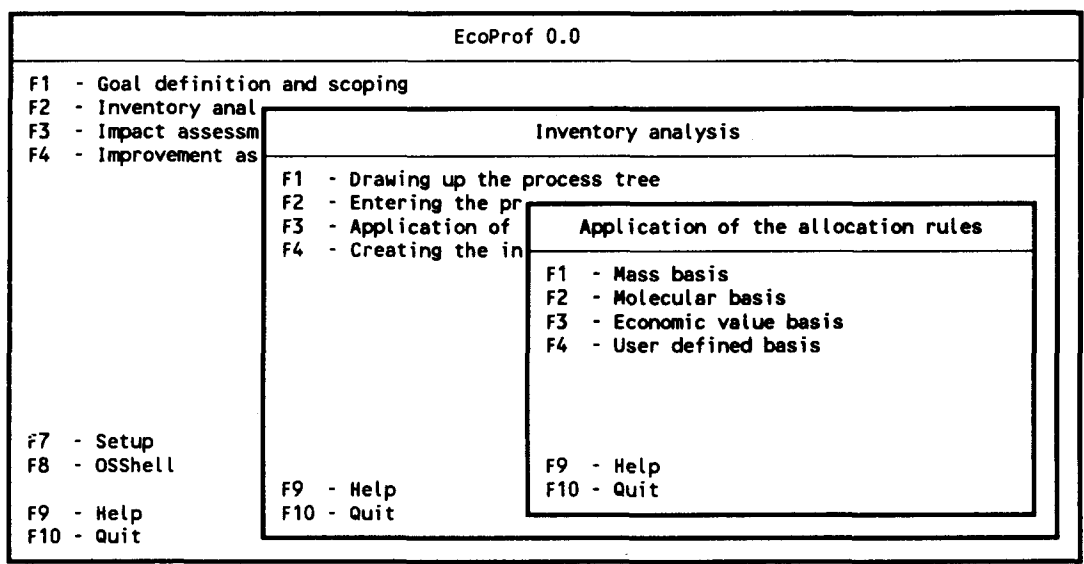


Figure 2 Example of the implementation of LCA in software following a rigid protocol

A clear distinction between the following categories is a basic requirement²:

- unknown;
- unknown but surely present;
- below detection limit;
- known to be zero.

Some of the problems associated with data on characterization and valuation are a matter of theory; in fact recipes such as ‘multiply with the global warming potential’ are dependent on the definition of these potentials and hence on the actual determined values.

Evaluation data are the most political part of LCA. The above-mentioned ‘recipe’ argument naturally applies. More important is the mutual influence of weighting data and theory. The theory does not produce weighting factors but helps to formulate the right questions to be answered by the experts or politicians. One should not ask ‘how important is global warming in relation to acidification?’, but

unambiguously specify what is meant by global warming, which method is used to characterize it, whether it is related to the world’s annual contribution, and much more⁵.

Conclusion

The influence of the disciplinary craftsmanship of computer specialists on methodology for LCA, as well as on daily practice, have been addressed. As one may not presume any implicit LCA knowledge from computers and hardly any from computer specialists, everything has to be formulated explicitly and unambiguously in a transparent way. This offers the most efficient opportunities to obtain computer models that do the right things. Sloppy questions yield fuzzy answers.

An advantage beyond the technical requirement when involving computer specialists is facilitation of communication among different schools of LCA

practitioners, among different schools of LCA methodologists, and between practitioners and methodologists. Discussions can be much more to-the-point and can yield more fruitful results.

A last argument is presented by methodology and data themselves: new insights can be reached by constructing a rigid protocol of steps with sharply defined activities. An example of this is the development of a method for improvement assessment, as a logical result of an attempt to give explicit formulas to compute the inventory table¹⁸.

The implementation in software of a transparent and explicit methodology for LCA is straightforward and relatively easy. It is our hope that programs for LCA will in the near future imitate the procedural structure as described in the methodological reports. *Figure 2* gives an example of such a program, based on the protocol of *Table 1*.

Evidently, reports of case studies gain clarity by following the same structure. If, for example, one wishes to study the methods of allocation of multiple processes encountered in various case studies, it would be an advantage if every report gave the details in the third step of the second component, so in § 2.3.

Uniformity sometimes seems very dull. In a world of increasing complexity in which environmental problems are challenging mankind, environmental, microeconomic and macroeconomic arguments must be judged independently on their relevance. Clarity of facts and opinions gives the only clue to the search for information needed to manage today's environmental problems efficiently.

References

- 1 Consoli, F., Allen, D., Boustead, I., Fava, J., Franklin, W., Jensen, A.A., de Oude, N., Parrish, R., Perriman, R., Postlethwaite, D., Quay, B., Séguin, J. and Vigon, B. 'Guidelines for Life-cycle Assessment: A "Code of Practice"', SETAC, Brussels/Pensacola, 1993
- 2 Fava, J.A., Denison, R., Jones, B., Curran, M.A., Vigon, B., Selke, S. and Barnum, J. (Eds) 'A Technical Framework for Life-cycle Assessment', SETAC, Washington, 1991
- 3 Fraunhofer-Institut für Lebensmitteltechnologie und Verpackung, Gesellschaft für Verpackungsmarktforschung, Institut für Energie- und Umweltforschung, 'Umweltprofile von Packstoffen und Packmitteln, Methode (Entwurf)', ILV, München, 1991
- 4 'Product Life Cycle Assessment—Principles and Methodology', Nord, Copenhagen/Stockholm, 1992
- 5 Heijungs, R., Guinée, J.B., Huppes, G., Lankreijer, R.M., Udo de Haes, H.A., Wegener Sleeswijk, A., Ansems, A.M.M., Eggels, P.G., van Duin, R. and de Goede, H.P. 'Environmental Life Cycle Assessment of Products. I: Guide—October 1992 II: Backgrounds—October 1992', CML, Leiden, 1992
- 6 Vigon, B.W., Tolle, D.A., Cornaby, W., Latham, H.C., Harrison, C.L., Boguski, T.L., Hunt, R.G. and Sellers, J.D. 'Product Life-cycle Assessment: Inventory Guidelines and Principles', EPA, Cincinnati, 1992
- 7 Goedkoop, M.J. 'SIMAPRO manual. Computer Programme for the Assessment of the Environmental Impacts of Products', PRé, Amersfoort, 1991
- 8 Lübker, B., Virtanen, Y., Mühlberger, M., Ingman, J., Vallance, B. and Alber, S. 'Life-cycle Analysis. IDEA, an International Database for Ecoprofile Analysis. A Tool for Decision Makers. Working paper, September 1991', IIASA, Laxenburg, 1991
- 9 'LCA Inventory Tool. Demo version. User manual', Chalmers Industriteknik, Göteborg, 1992
- 10 Boustead, I. and Hancock, G.F. 'EEC Directive 85/339, UK Data 1986, a Report for INCPEN', The Open University, 1989
- 11 Habersatter, K. 'Ökobilanzen von Packstoffen. Stand 1990', BUWAL, Bern, 1991
- 12 Boustead, I. 'Eco-profiles', PWMI, Brussels, 1993
- 13 Pedersen, B. and Christiansen, K. in 'Product Life Cycle Assessment—Principles and Methodology', Nord, Copenhagen/Stockholm, 1992
- 14 Guinée, J.B., Huppes, G. and Udo de Haes, H.A. *J. Cleaner. Prod.* 1993, 1, 3
- 15 'Critical Review of LCA Practice', ENDS report 219, 1993, pp 25–26
- 16 'Life-cycle Assessment. Workshop Report', SETAC, Brussels, 1992
- 17 Huppes, G. in 'Life-cycle Assessment', SETAC, Brussels, 1992
- 18 Heijungs, R. *Ecol. Econ.* in press